An Overview of ASC Efforts in Parallel First-Order Sn Methods

S.D. Pautz,*

*Sandia National Laboratory, Albuquerque, New Mexico, 87185

As part of the Department of Energy's Accelerated Strategic Computing Initiative (ASCI – now ASC) efforts were made to improve the algorithms used in various multiphysics codes. In the area of radiation transport there were multiple efforts to improve first-order discrete ordinates (Sn) methods for parallel computing with unstructured meshes. We summarize the efforts to date.

Introduction

The cessation of nuclear testing in the United States during the 1990's created an increased need for improvements in other tools used to maintain the stockpile. The Accelerated Strategic Computing Initiative (ASCI) was created to address a broad range of computational challenges at the U.S. weapons laboratories, encompassing both hardware and software. Algorithmic improvements were sought in a variety of computational disciplines, including radiation transport. One transport approach targeted for improvements is the method of discrete ordinates (S_n) as applied to the first-order form of the Boltzmann transport equation, in particular for parallel computations on unstructured meshes. The purpose of this paper is to survey the research into parallel first-order S_n methods performed within the ASCI program.

Background

In this section we briefly describe some of the important characteristics of first-order S_n methods. We will concentrate on those aspects that affect parallelization approaches, especially where they differ from other computational disciplines. More details are available elsewhere (Pautz, 2002).

Serial sweeps on structured meshes

A fundamental process used by first-order S_n methods is the method of sweeping, in which incident angular fluxes and internal sources are traced along a direction of travel through a spatial domain. We depict a typical situation in Figures 1 and 2 for a structured mesh. For each direction of particle travel $\mathbf{\Omega}$ the iteration "sweeps" through the mesh, beginning with "upstream" spatial cells (in Figure 1, those in the top right corner) and proceeding through "downstream" cells (in this case, those towards the bottom left of the mesh). In each cell the transport algorithm approximately inverts the "streaming-plus-collision" operator; solutions within upstream cells are necessary for the calculation

within cells immediately downstream. The sweep process continues until all cells have been solved; particles either will have been removed from the process through collisions, or uncollided fluxes will exit the downstream side of the spatial domain. Usually sweeps occur within the context of "source iterations": multiple sweeps are used, with a recalculation of scattering sources after each set of sweeps. The sweeps and source calculations are repeated until both the streaming and source terms have converged.

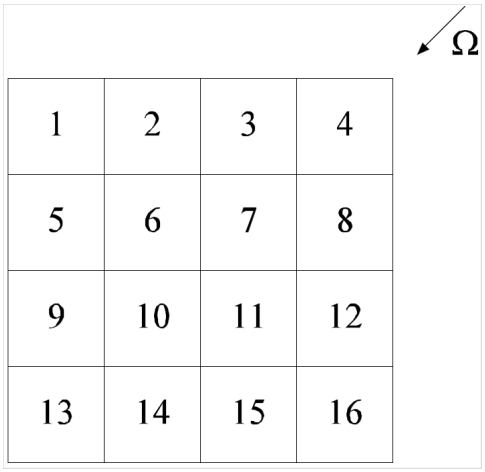

Figure 1. Structured mesh sweep.

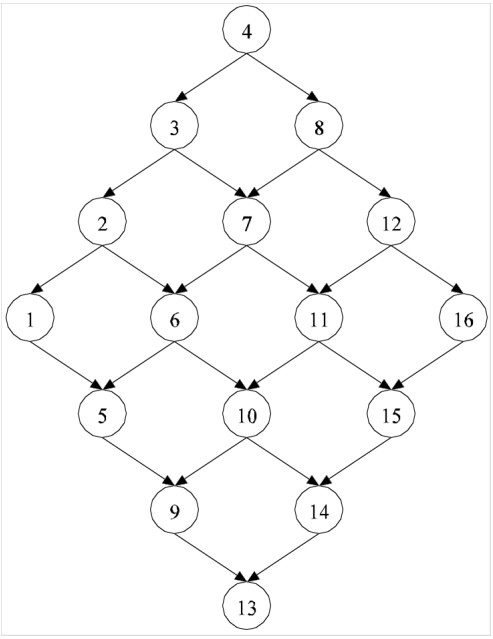

Figure 2. Dependency graph for structured mesh sweep.

The geometric situation shown in Figure 1 may be described by the directed acyclic graph (DAG) in Figure 2; each cell is mapped onto a graph vertex, and each shared face is mapped onto a graph edge. There is a DAG for each direction within the angular

quadrature set used by the transport calculation. As Figure 2 implies, the sweeping algorithm is not free to solve the cells in any order if it wishes to respect the dependency graph. Violations of the dependency graph force the sweep algorithm to use incoming angular fluxes obtained via other means, usually from previous sweep iterations ("previous iterate" data). Violations of this sort can degrade the convergence rate of the iterations, depending on the nature of the transport problem. For serial calculations it is easy to respect the dependency graph; one may simply solve all the cells in one level of the DAG (in any order) before proceeding to lower levels. This "sweep ordering" is a simple matter of bookkeeping for serial sweeps on unstructured meshes, but it is an additional constraint not seen in many other iterative methods, which often are insensitive to the ordering of tasks within the innermost algorithmic loops.

Serial sweeps on unstructured meshes

In Figures 3 and 4 we depict the sweep problem for a representative unstructured mesh. As in the structured case we obtain a directed graph for each angle; unlike the structured case we find that the sweep graphs for unstructured meshes may be directed cyclic graphs (DCGs). One could attempt to simultaneously solve for all the cells in a cycle (i.e. treat them as a single vertex in the graph), but this can be complicated and expensive. A simpler approach is to treat the DCG as a DAG by "breaking" the cycles; one may ignore some of the dependencies in the cycle by using previous iterate information. As discussed above this may degrade the iterative convergence rate. Furthermore the directed graphs produced by unstructured meshes are more complicated; they usually cannot be analyzed by inspection to yield simple ordering expressions. Therefore the sweeping of unstructured meshes, even in serial calculations, in general presents the following challenges not seen with structured meshes:

- DCGs must be converted to DAGs.
- The above process may degrade the effectiveness of the iterations.
- The sweep ordering requires more general bookkeeping.

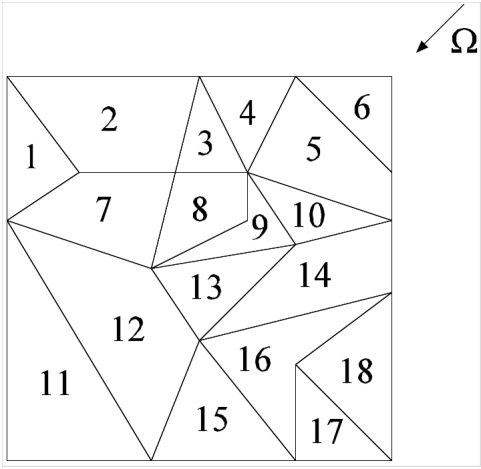

Figure 3. Unstructured mesh sweep.

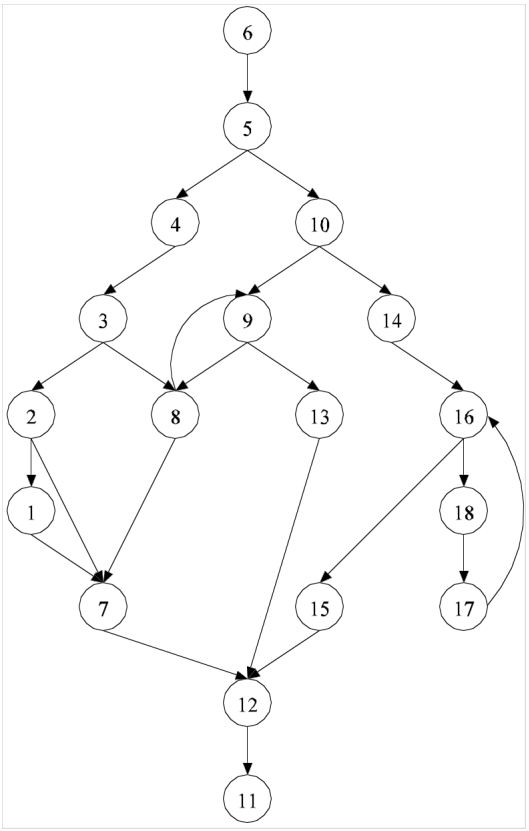

Figure 4. Dependency graph for unstructured mesh sweep.

Parallel sweeps on structured meshes

In order to perform any type of parallel computations it is necessary to have a strategy for distributing the tasks to the various processors. In transport calculations we may in general decompose the tasks in the energy, angle, and/or spatial variables. Except for specialized applications there are strong arguments in favor of using a spatial domain decomposition alone; most research has assumed that this is the case.

In many fields of scientific computing algorithms have been developed to distribute the spatial mesh; such decompositions typically attempt to minimize the communication volume and balance the overall workload. For sweep-based algorithms, however, we note that we must analyze such decompositions in the context of the sweep graphs. Faithful adherence to the dependency graph may result in idle processors if all of their assigned tasks are clustered in only a few levels of the sweep graph, thus degrading the efficiency of the sweep. Unfortunately the spatial decompositions that are produced by optimizing just the communication volume and the overall workload (which is true of all commonly available partitioners) have been shown to be far from optimal for sweep-based algorithms; one cannot even construct a scalable sweep algorithm with these decompositions.

The nature of the sweep graph affects communication costs as well. Edges in the sweep graph represent data flow; along partition boundaries such edges are associated with a message. Naïve implementation of the sweep process will yield many tiny

messages, which are generally more expensive than a few large messages of the same total volume. For elliptic operators it is common to minimize the number of messages by using a block-Jacobi iteration in which each processor performs work in every local spatial cell, followed by a communication step in which all partition boundary data is exchanged in a small number of large messages. For the hyperbolic sweep process such an approach grossly violates the sweep graph, yielding in some cases a greatly degraded iterative method. One may also maintain large message sizes by requiring processors to wait until all upstream data is available (if that is even possible), but this can quickly degrade the efficiency of the sweeps by producing idle processors. Somehow these concerns must all be balanced for a sweep process to be both effective and efficient.

Fortunately the KBA algorithm has been developed for parallel sweeping of structured meshes (Baker and Koch, 1998). It consists of a special domain decomposition approach (shown in Figure 5) that results in tasks from many different levels of the sweep graph being assigned to each processor. The KBA algorithm also specifies a method for ordering the tasks such that processors are idle for only a small fraction of the time, no violations of the sweep graph occur, and messages are not too frequent or small. It does this in part by sweeping a "chunk" or group of cells at a time on each processor in between communication steps; the number of cells per chunk, and consequently the number of chunks per processor, is optimized to avoid the communication costs associated with small messages and excessive processor idling associated with infrequent messages. KBA and its derivatives have been found to be very efficient for parallel transport on structured meshes.

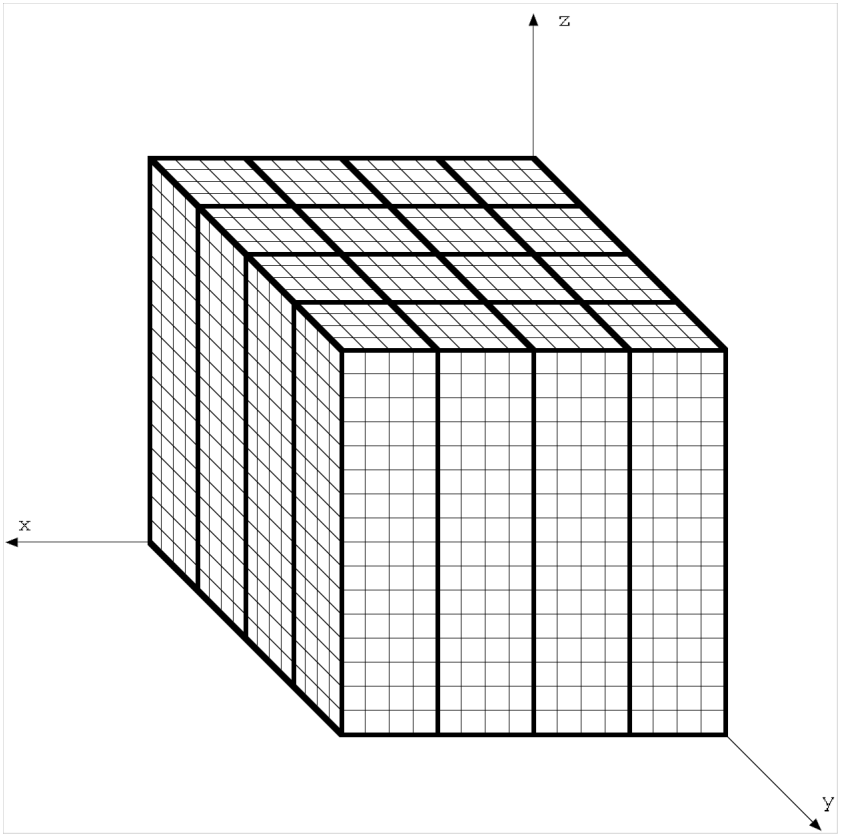

Figure 5. KBA decomposition of structured mesh.

In summary, parallel transport on structured meshes introduces the following challenges compared to serial calculations, some of which conflict with each other:

- Spatial decompositions should avoid large total communication costs.
- Spatial decompositions should avoid task-to-processor assignments that are unevenly distributed across the sweep graph levels.
- The sweep ordering scheme should avoid excessive violations of the sweep graph.
- The ordering scheme should avoid large communication costs associated with small message sizes.
- The ordering scheme should avoid processor idling associated with large message sizes.

The KBA approach has effectively addressed all of the above concerns

Parallel sweeps on unstructured meshes

"The partitioning and scheduling problems are vastly more difficult on arbitrary grids (Adams et al., 2002)." Parallel transport on unstructured meshes must deal with the same issues seen in parallel transport on structured meshes or serial transport on unstructured meshes, but they are more complicated. For example, conversion of DCGs to DAGs now generally must occur in parallel, but the usual (serial) algorithms to do so are not readily parallelized. Parallel unstructured mesh sweeps also must deal with issues not seen in other contexts. For example, grouping cells into chunks (also called cellsets) as in KBA will generally convert even an acyclic cell dependency graph into a cellset dependency graph with many cycles. Algorithmic approaches designed to address one set of issues usually conflict with other issues, yielding a highly coupled optimization problem. The problem of finding an optimal approach is NP-complete, so any proposed algorithms will be heuristic in nature.

Some of the questions facing a researcher in this field are the following:

- How should the mesh be distributed?
- How can DCGs be efficiently converted to DAGs?
- In what order should the distributed sweep graph be traversed?
- When should communication occur?
- How many violations of the sweep graph can be tolerated?
- How can one precondition the resulting iterative system?

The effectiveness of various proposed algorithms will depend on the spatial meshes used, the spatial discretization on those meshes, the angular quadrature, the energy discretization, the transport regime of the application, the computer architecture, and perhaps other factors.

Research in Parallel First-Order S_n Methods

The bulk of the ASCI research into parallel sweep-based transport has occurred in two contexts. There has been a multiyear effort at Texas A&M University (TAMU) to address this topic in its full generality. There also have been focused efforts at each of the three U.S. weapons laboratories (the "tri-labs"). These efforts have not been entirely separate, but have influenced each other in various ways.

Texas A&M University research

A major effort to understand and improve the algorithms for parallel first-order S_n transport was initiated in 1998 at Texas A&M University by means of funding from the Academic Strategic Alliances Program (ASAP) of ASCI. Direct funding by the tri-labs beginning in 2002 has allowed the research program to continue beyond its original ASAP contract. The effort consists of a collaboration between the Departments of Nuclear Engineering and Computer Science, involving several faculty members and numerous students. Researchers from the tri-labs have continuously interacted with this effort, primarily through numerous workshops at TAMU but also through joint research projects and by internships of TAMU students at the labs (Adams et al., 2002).

The TAMU effort has resulted in much greater understanding of the problem, with some concrete results already proven and other algorithmic ideas being explored. Full details are available in their reports and workshop proceedings (http://parasol.tamu.edu/asci); we report here some of the important results.

One of the first main insights of the effort is that the parallel sweeping problem may be described as a special case of the general scheduling problem from computer science, with constraints. The main constraint is the desire to decompose only in the spatial variable. Without that constraint one could apply a variety of existing scheduling algorithms to the sweep graphs for each angle in the quadrature set, yielding a decomposition in both angle and space (and perhaps energy as well). The existence of this constraint invalidates the direct use of such algorithms, but not necessarily the insights developed over many years by the scheduling theory community.

Another contribution made by the TAMU effort is the creation of a conceptual framework to describe the sweep process; it is general enough to encompass a variety of existing or proposed sweep algorithms. The classical (serial) description of sweep-based transport iterations consists of a series of nested loops; the order of the nesting has long been fixed by the physics of many important transport problems:

```
Iteration loop on group-to-group upscattering
    Loop over energy groups (highest- to lowest-energy)
        Iteration loop on within-group scattering
            Loop over angles in this group
                Loop over cells in appropriate order for this angle
```

The TAMU effort generalized this by allowing individual energy groups, angles, and cells to be organized into groupsets, anglesets, and cellsets, thereby splitting the nested loops into several more nested loops:

```
Iteration loop on groupset-to-groupset upscattering
    Loop over energy groupsets (order selected by user or by learning algorithm)
        Iteration loop on within-groupset scattering
```

Loop over (angleset, cellset) pairs in this groupset
  Loop over cells in cellset in appropriate order for this angleset
    Loop over angles in this angleset
      Loop over groups in this groupset

The above iterative framework describes sweep algorithms that at least partially reverse the order of loop nesting for the sake of scheduling concerns and cache performance as well as the classic sweep algorithm. This allows one to more readily compare competing sweep algorithms. It also allows a code written in the generalized form to more readily switch between different sweep algorithms, perhaps dynamically.

The TAMU effort has done extensive analysis of the KBA algorithm in both 2D and 3D, for the sake of improved algorithms for structured meshes and to gain insights into unstructured mesh sweeps. They have found that although the KBA approach is good, there are extensions to it that are somewhat better. In particular they have been able to express the KBA decomposition strategy in a generalized parametric form; good parameter choices are made by solving a small optimization problem to minimize the solution time.

The TAMU effort has also studied sweep-based preconditioners. In the past much effort had been devoted to a class of preconditioners called diffusion synthetic acceleration (DSA) (Adams and Larsen, 2002). Diffusion-based preconditioners are attractive because the slowest converging error modes of source iteration are often diffusive in nature and may be effectively dampened by DSA. Unfortunately it is difficult to construct DSA schemes that are both effective (reduce the iteration count) and efficient (easily solved), especially as more sophisticated transport discretizations are created. Diffusion equations also require a different solution methodology than sweep-based transport, adding complications to code implementation and parallelization strategy. A class of sweep-based preconditioners called transport synthetic acceleration (TSA) has been developed to address these issues. TSA methods use the same operator form, albeit of a lower order, as the high level transport operator; such preconditioners immediately benefit from algorithmic advances in sweep-based transport and may reuse the sweep coding. TAMU researchers have analyzed TSA schemes in the context of various approaches to the high-level sweep scheduling.

The final TAMU contribution we wish to discuss is the development of a code library that is useful to both transport and non-transport applications. Researchers there are in the process of developing the Standard Template Adaptive Parallel Library (STAPL), which is an extension of the C++ STL. The STAPL library adds parallelism "under the hood" to hide the underlying communication protocol. It also adapts itself to both the local architecture and to runtime information. It is hoped that codes based on this library will be more efficient, more clearly written and maintainable, and also more adaptable to changing computational environments. An initial public release of STAPL is expected within the next year (Rauchwerger, 2005).

Tri-lab research

Lawrence Livermore National Laboratory (LLNL)

Work on parallel first-order S_n methods on polyhedral meshes has been performed within the TETON code. TETON uses a block-Jacobi sweep process; in the TAMU nomenclature this is equivalent to one cellset per processor, with complete violation of

the sweep graph along partition boundaries during the sweep itself. TETON performs threading over angle within shared-memory nodes, where there are one or more processors per node. The code uses "energy batching" (groupsets); the threading and energy batching are intended to improve cache performance. The LLNL researchers also have been extensively involved in TSA research (Nowak and Nemanic, 1999).

Recent work has improved the block-Jacobi iteration by allowing the use of new iterative information where possible (fewer violations of the sweep graph along partition boundaries), provided that processors are not required to wait for new information. Livermore has also been successful in developing a "stretched-and-filtered" TSA that is much more effective than earlier TSA methods (Nowak, 2005).

Los Alamos National Laboratory (LANL)

The original KBA work on structured meshes has continued within the PARTISN code, including its extensions to AMR meshes (Baker, 2001). PARTISN has also implemented TSA preconditioners as an alternative to its existing DSA method (Adams et al., 2002).

The research code Tycho was developed to examine different ordering algorithms for parallel sweeps on tetrahedral meshes. Unlike TETON it rigorously respects the dependency graph; in TAMU terminology it uses groupset/angleset/cellset sizes of unity, with the caveat that it delays communications by concatenating messages in a manner similar to larger cellsets. This research examined the practical limitations of standard mesh partitioning strategies. The Tycho research determined that reasonable parallel efficiencies could be obtained with standard decompositions on dozens or hundreds of processors, depending on the angular quadrature order (Pautz, 2002).

Another separate effort at LANL focused on performance modeling of both structured and unstructured mesh sweeps. This work yielded insights and improvements related to communication patterns during the sweeps (Mathis and Kerbyson, 2005).

More recently the Capsaicin project has been initiated at LANL to implement parallel first-order S_n technologies for thermal radiation transport. The project has implemented several sweep algorithms ranging from block-Jacobi to full (graph-respecting) sweeps (Thompson, 2005).

Sandia National Laboratories (SNL)

The original ASCI work at Sandia focused on mesh partitioning and parallel cycle detection. Sandia researchers produced a geometry-based mesh partitioner that attempted to produce KBA-like decompositions for unstructured meshes; their studies showed improved sweep performance characteristics. In conjunction with TAMU they also created a sweep cycle detection algorithm that effectively parallelized a breadth-first search algorithm (Plimpton et al., 2000).

Currently Sandia researchers are developing a new first-order S_n code within the Ceptre effort intended for neutral and possibly charged particles. It makes use of the parallel sweep cycle code developed earlier at Sandia.

Conclusions

Creating effective and efficient parallel transport sweep methods is an important but difficult task. A substantial amount of research has been conducted within the ASCI

program to address this issue. This research has greatly improved our understanding of the problem and resulted in several improved algorithms. Ongoing research into other promising approaches continues.

Acknowledgements

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DEAC04-94AL85000.

References

Adams, M.L., Amato, N., Nelson, P., and Rauchwerger, L., "Efficient Massively-Parallel Implementation of Modern Deterministic Transport Calculations," Texas A&M University, College Station, TX (2002).

Adams, M.L. and Larsen, E.W., "Fast Iterative Methods for Discrete-Ordinates Particle Transport Calculations," *Progress in Nuclear Energy*, **40**, No. 1, 3-159 (2002).

Baker, R.S. and Koch, K.R., "An S_n Algorithm for the Massively Parallel CM-200 Computer," *Nucl. Sci. Eng.*, **128**, 312 (1998).

Baker, R.S., Los Alamos National Laboratory, Los Alamos, private communication (2001).

Mathis, M.M. and Kerbyson, D.J., "A General Performance Model of Structured and Unstructured Mesh Particle Transport Computations," *J. Supercomputing*, To Appear (2005).

Nowak, P.F. and Nemanic, M.K., "Radiation Transport Calculations on Unstructured Grids Using a Spatially Decomposed and Threaded Algorithm," *Proc. Int. Conf. Mathematics and Computations, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid, Spain, September 27-30, 1999, Vol. 1, 379-390 (1999).

Nowak, P.F., Lawrence Livermore National Laboratory, Livermore, private communication (2005).

Pautz, S.D., "An Algorithm for Parallel S_n Sweeps on Unstructured Meshes," *Nucl. Sci. Eng.*, **140**, 111-136 (2002).

Plimpton, S., Hendrickson, B., Burns, S., and McLendon, W. III, "Parallel Algorithms for Radiation Transport on Unstructured Grids," *Proc. SuperComputing 2000*, Dallas, TX November 4-10, 2000.

Rauchwerger, L., Texas A&M University, College Station, TX, private communication (2005).

Thompson, K.G., Los Alamos National Laboratory, Los Alamos, private communication (2005).